BKLM GCR2

Yinghui GUAN

Main problem

efficiency matching with CDC
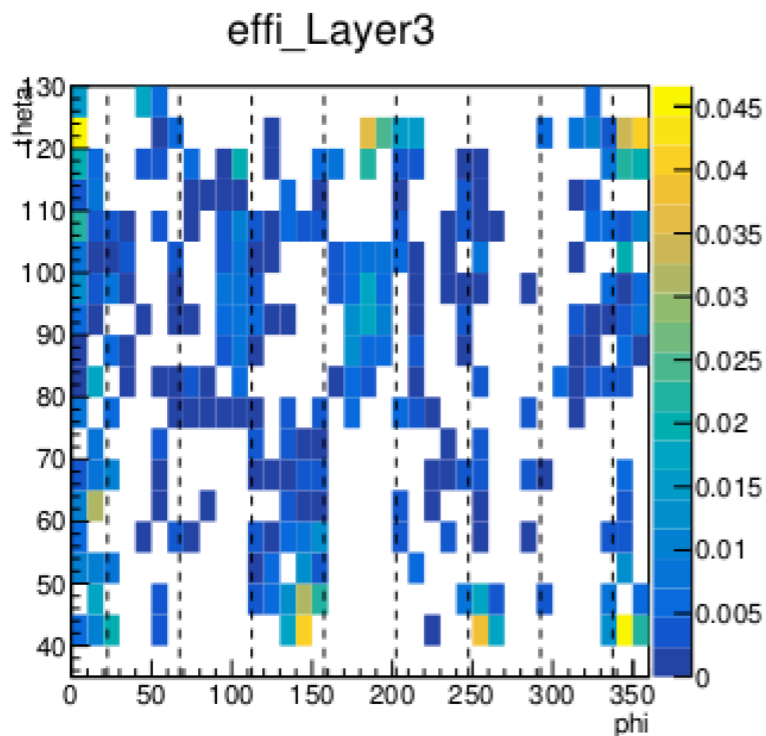
effi_Layer3

good run

run configuration SAME!
RPC look back window SAME!

effi_Layer3

bad run

track rate, not efficiency

from Zach. Stottler

| Run Number	Events Sampled	% of Events with KLM Tracks
00286	14.9k	77.1%
00287	14.9k	0.6%
00288	15.0k	77.2%
00289	15.0k	0.5%
00290	14.9k	77.7%
00291	14.9k	77.5%
00292	15.0k	0.5%
00295	14.9k	77.6%
00296	14.9k	0.5%
00297	14.9k	77.6%
00298	14.9k	77.7%
00299	14.9k	0.7%
00300	14.9k	77.6%
00301	14.9k	0.5%
00303	14.9k	77.7%
00304	14.9k	77.6%
00305	14.9k	77.5%
00306	14.9k	77.6%
00307	15.0k	0.5%
00309	14.9k	78.0%
00310	14.9k	0.5%
00311	14.9k	77.7%
00312	15.0k	0.6%
00313	14.9k	77.4%

2

number of total digits.

track go through one sector

track go through two sectors
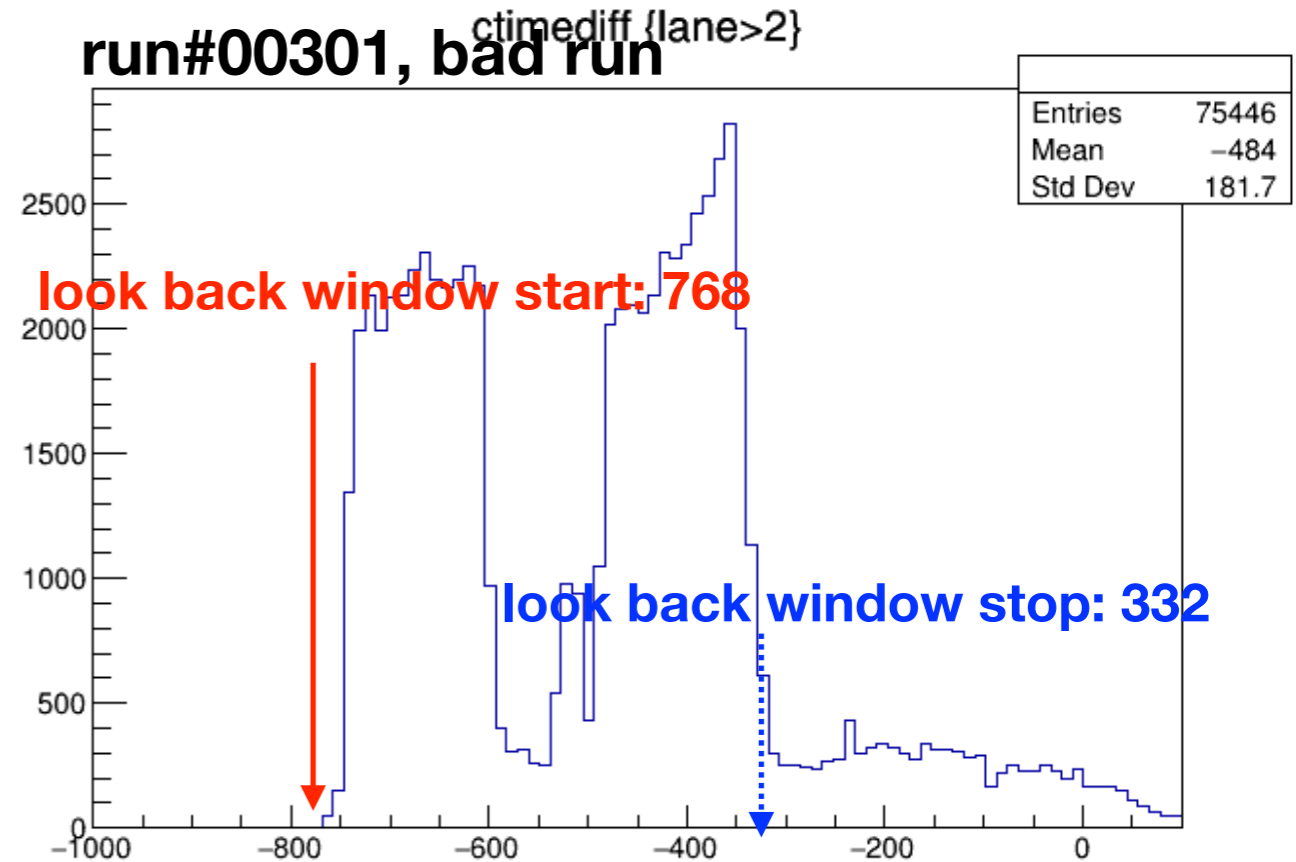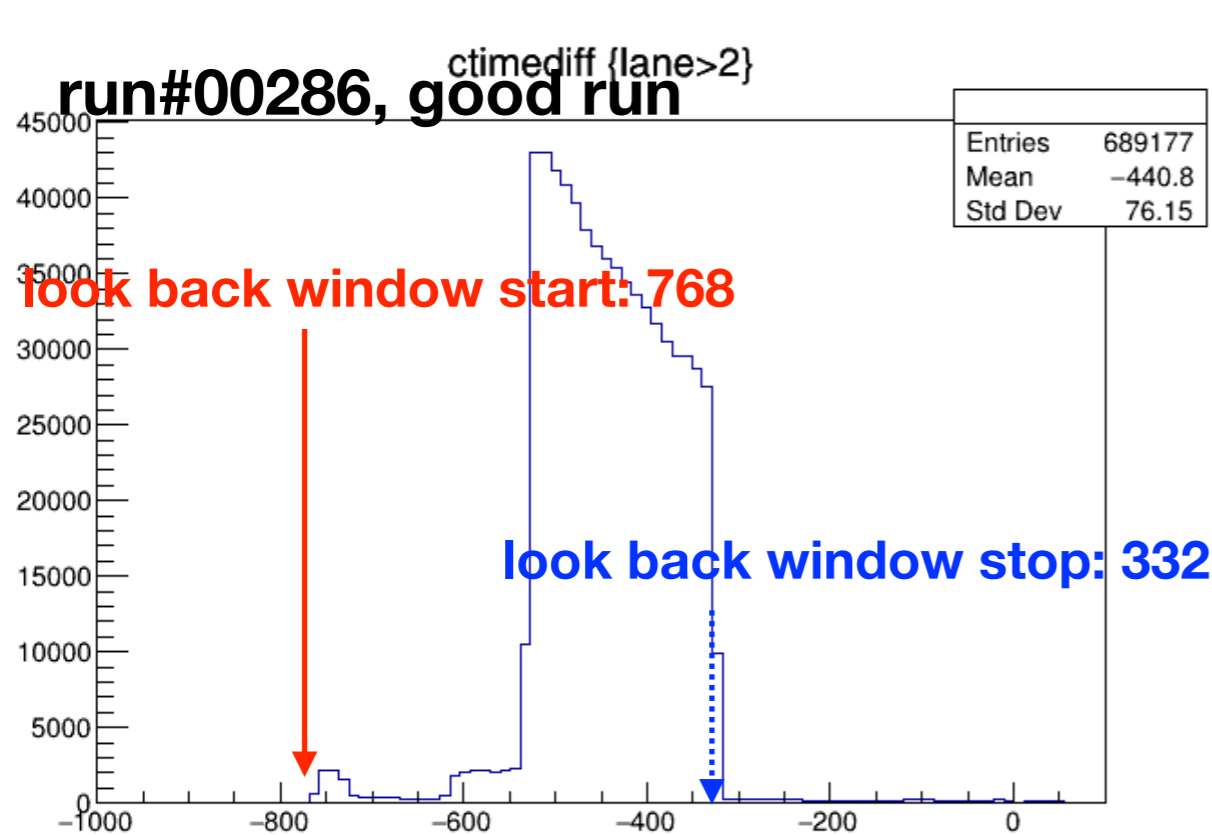

run#00286, good run

run#00301, bad run
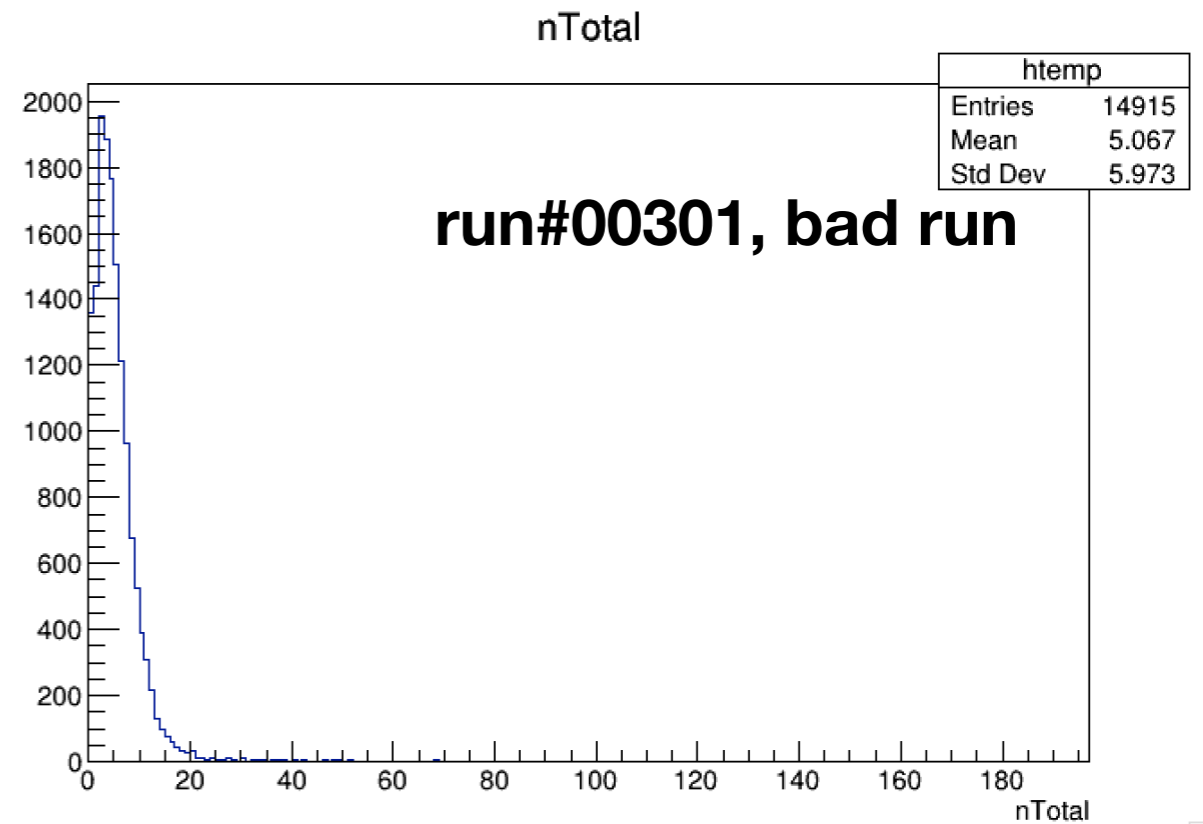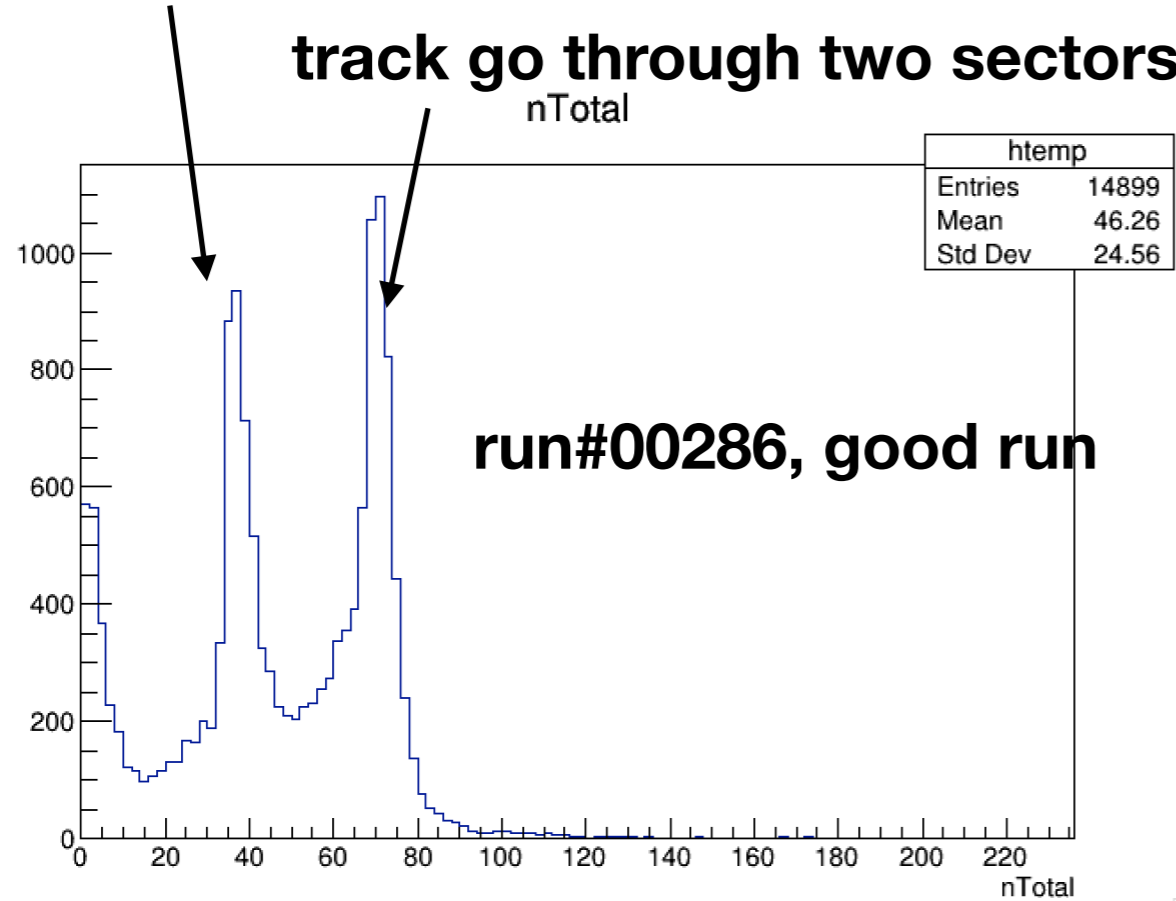
no any cutoff.

CTIME w.r.t trigger CTIME

Look at all KLM hits (no any cutoff). Look at CTIME differences from trigger CTIME.

run#00286, good run

ctimediff {lane>2}

Entries	689177
Mean	−440.8
Std Dev	76.15

look back window start: 768

look back window stop: 332

run#00301, bad run

ctimediff {lane>2}

Entries	75446
Mean	−484
Std Dev	181.7

look back window start: 768

look back window stop: 332
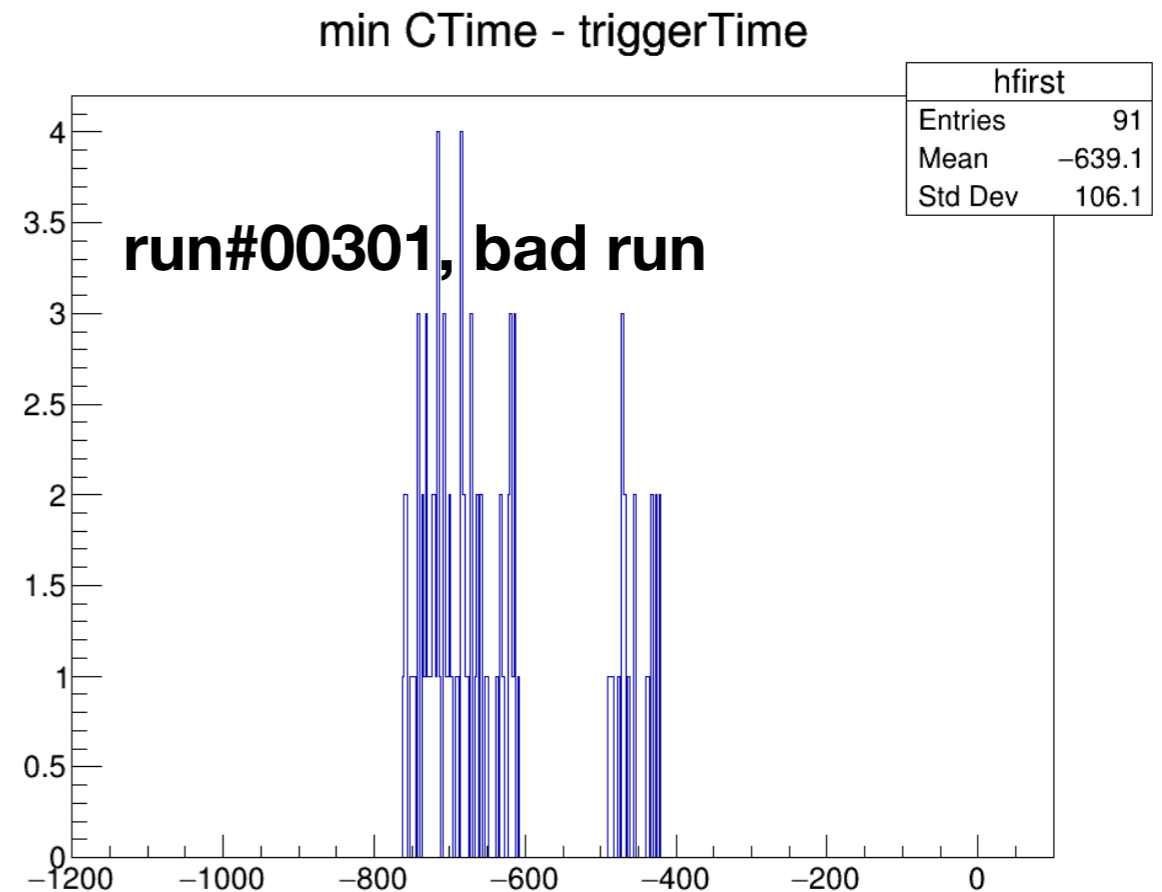
- **Our RPC look back window [768, 332], [6.1us - 2.7us], in hex-decimal: [0x300, 0x14C].**
- **CTIME of KLM hits increases during read out process, so due to readout delay, the largest CTIME are not consistent with triggerCTIME - window stop. But the minimum CTIME is consistent with triggerCTIME - window start (see above plots).**
- **We can double check the look back window from this CTIME distribution.**

CTIME w.r.t Trigger CTIME

Look at hits on KLM stand alone tracks, take the earliest hit (minimum CTIME). Look at its differences from trigger CTIME.

min CTime - triggerTime

hfirst	
Entries	10978
Mean	−524.3
Std Dev	15.4

run#00286, good run

8ns
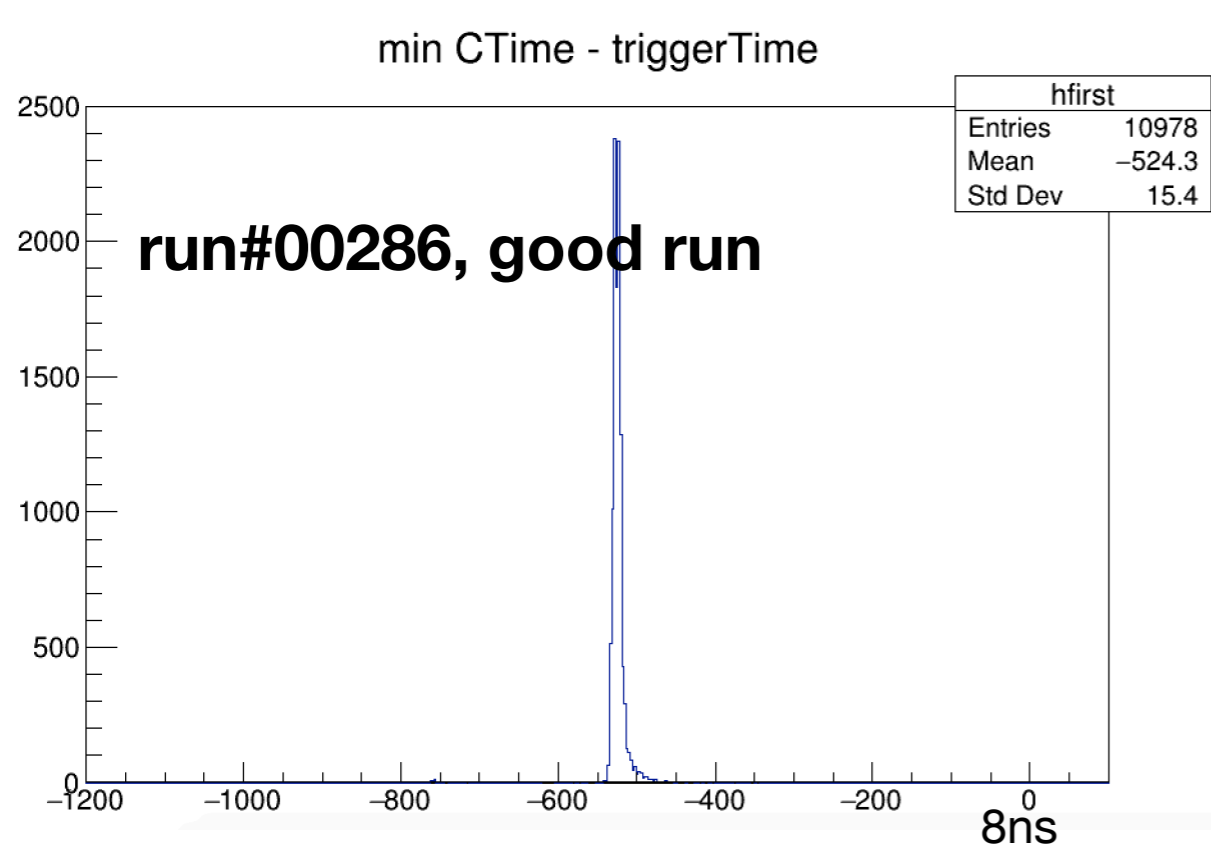
it means trigger latency ~ 530*8ns ~ 4.2us

min CTime - triggerTime

hfirst	
Entries	91
Mean	−639.1
Std Dev	106.1

run#00301, bad run

Our RPC look back window setting is : [6.1us - 2.7us], should be ok.
L1 latency is stable according to CDC.

GUI slow control SW

- First tried, adding a sleep(30) after sending slow control data in LOAD().

- The next day(Feb.22), commented out all sending slow control data (also sleep(30)). On the other hand, re-sending slow control overwrites Isar's calibration using command line. The GUI for KLM part does nothing but read a few DC registers (RPC loopback window), not any write any register. After this, we still have the efficiency problem.

- But..(the next page)

run list

Feb.25 (Sunday)

Run Number	Events Sampled	% of Events with KLM Tracks
00942	6.6k	60.0%
00943	6.7k	60.2%
00944	6.7k	0.5%
00945	6.7k	0.5%
00949	6.7k	60.0%
00951	6.6k	58.3%
00952	6.6k	0.4%
00953	6.6k	58.2%
00954	6.7k	58.9%
00955	6.7k	60.0%
00956	10.0k	59.6%
00957	10.0k	59.4%
00958	10.0k	60.1%
00959	10.0k	58.9%
00960	10.0k	59.2%
00961	10.0k	59.6%
00962	10.0k	59.0%

Feb.26 (Monday)

still have efficiency problem by looking at first runs, haven't checked all runs yet.